

رمز المقرر: ITGS211.....
 الفصل الدراسي: خريف 2021-2022
 الزمن: 120 د



جامعة طرابلس
 كلية تقنية المعلومات
 الامتحان النهائي (نظري)

المقرر: البرمجة الشيئية (باستخدام Java) **الإجابة** اجمالي الدرجات: ... 50 درجة

الإجابة: بقلم الحبر وبخط واضح، ولكل سؤال إجابة واحدة وواحدة فقط (أجب عن جميع الأسئلة)
 السؤال الأول: (6 درجات لكل فقرة 2 درجتان)

النتائج	Code
<p>x.no= 33 z.no= 3 class Two</p>	<p>1 class One{ public int no; public One(int m){ this.no=m; } public One(int a, int b){ this.no=a/b; } } class Two extends One{ public Two(){ super(15,4); } public void printT(){ System.out.println("class Two"); } } public class JavaApplication12 { public static void main(String[] args) { One x= new One(33); Two z= new Two(); System.out.println("x.no= "+x.no); System.out.println("z.no= "+z.no); z.printT(); } }</p>
<p>50 10 10 10 10</p>	<p>2 public static void mystery(int[] array) { int tmp = array[array.length - 1]; for (int i = 1; i < array.length; i++) { array[i] = array[i - 1]; } array[0] = tmp; } public static void main(String[] args) { int[] list = { 10, 20, 30, 40, 50 }; mystery(list); for (int i = 0; i < list.length; i++) { System.out.print(list[i]+ " "); } }</p>

3	<pre> char x = 'A'; while(x != 'D'){ switch(x){ case 'A': System.out.println(x); x = 'D'; case 'B': System.out.println(x); x = 'C'; break; case 'C': System.out.println(x); x = 'D'; default: continue; } } </pre>	<p style="text-align: center;">A D C</p>
---	--	--

السؤال الثاني (10 درجات):

ضع حرف (T) في حال أن الجملة صحيحة و حرف (F) في حال أن الجملة خاطئة.

1. أي صنف يحتوي على دالة مجردة هو صنف مجرد (F).
2. يجب إسناد قيمة لحقل من النوع ثابت (final) عند النقطة التي تم فيها الإعلان عنه داخل صنف (T).
3. لا يمكن للصنف أن يحتوي على دوال متعددة بأسماء متطابقة (F).
4. يجب أن تُرجع الدالة في Java قيمة واحدة (F).
5. يتم تنفيذ دالة البناء في Java مرة واحدة فقط ، عند إنشاء كائن لأول مرة (T).
6. لا يمكن إنشاء كائن من صنف مجرد (T).
7. يجب ألا تتم عملية تركيب الدوال في الأصناف الفرعية أبداً من الدوال الموجودة في الصنف الأصلي (F).
8. يشير تعدد الأشكال إلى حقيقة أن كائنات مختلفة يمكن أن تستجيب لنفس الطريقة بطرق مختلفة (T).
9. "this" هو متغير خاص في Java ، ويجب الإعلان عنه ، تجله Java متاحة تلقائياً في الدوال فقط (F).
10. الصنف (class) عبارة عن مصنع لإنشاء الكائنات (objects) (T).

السؤال الثالث: (8 درجات 4 للفقرة الأولى ، 4 للفقرة الثانية)

1. أوجد قيمة التعبيرات الآتية:

اعتبر أن قيمة `int x=9`

<code>x / 3.0 / 6 * 2</code>	1
<code>14 % (x / 5 + 1)</code>	0
<code>--x + ++x</code>	17
<code>--x == x++</code>	true

2. اعد كتابة الكود التالي باستخدام جملة **if-else** على اعتبار أن المتغير **grade** من النوع الحرفي (**char**).

```
switch (grade) {
    case 'A':
        System.out.println("Excellent");
    case 'B':
        System.out.println("Good");
    case 'C':
        System.out.println("OK");
    default:
        System.out.println("Let's talk");
}
```

```
if (grade == 'A') System.out.println("Excellent");
if (grade == 'A' || grade == 'B')
    System.out.println("Good");
if (grade == 'A' || grade == 'B' || grade == 'C')
    System.out.println("OK");
System.out.println("Let's talk");
```

السؤال الرابع: (10 درجات)

1. كون صنف سيمثل فريق كرة قدم، اجعل إمكانية الوصول لهذا الصنف باسم **FootballTeam** من أي حزمة (**package**) في البرنامج.
2. أضف الحقول للخصائص التالية، والتي لا يمكن الوصول إليها من خارج الصنف.
3. اكتب دالة البناء التي تستقبل اسم الفريق **Name** وعدد مرات الفوز **Wins** وعدد مرات الخسارة **losses** كمعاملات ويسند الخصائص للصنف لتلك القيم. إمكانية الوصول إلى دالة البناء من أي حزمة (**package**).
4. اكتب دالة بناء أخرى تأخذ اسم الفريق فقط كمعامل، يجب على دالة البناء تعيين اسم الفريق إلى القيمة الممررة من المعامل وتعيين عدد مرات الفوز (**Wins**) والخسارة (**losses**) إلى 0.
5. اكتب دوال (**sets**, **gets**) التي ترجع اسم الفريق وعدد الفوز وعدد الخسارة. يجب الوصول لهذه الدوال من أي حزمة في البرنامج (**package**).
6. اكتب بعد ذلك دالة لزيادة عدد مرات الفوز بمقدار 1 ودالة أخرى لزيادة عدد مرات الخسارة بمقدار واحد. يمكن الوصول لهذه الدوال من خارج الصنف **FootballTeam**.
7. اكتب دالة ترجع بقية **True** عندما يكون لدى الفريق "سجل جيد" (**good record**)، مما يعني أن الفريق يحقق فوز أكثر من الخسارة. يجب أن تكون هذه الدالة متاحة من أي حزمة (**package**).
8. أخيرًا، أضف الدالة الرئيسية، قم بتكوين كائن فريق كرة قدم يسمى **"AITI"** بثلاثة مرات فوز و 5 خسارة. قم باستدعاء الدالة التي تعود بقيمة **true** عندما يكون لدى الفريق سجل جيد وقم بطباعة النتيجة. الآن قم بإجراء ثلاث استدعاءات للدالة التي تزيد عدد مرات الفوز بمقدار 1. أخيرًا، قم باستدعاء الدالة "التسجيل الجيد" مرة أخرى وقم بطباعة النتيجة.

```

/** Question 1 (5 pts)
 *   -2 for no package statement
 *   -2 for no public keyword
 *   -1 for misspelled or mis-capitalized class or package name.
 */
package football;

public class FootballTeam {

    /** Question 2 (4 pts)
     *   -2 if any field is not private
     *   -2 if type of any field is wrong
     */
    private String name;
    private int wins;
    private int losses;

    /** Question 3 (4 pts)
     *   -1 missing public keyword
     *   -1 wrong parameter types or number
     *   -2 for incorrect body
     */
    public FootballTeam (String name, int wins, int losses) {
        this.name = name;
        this.wins = wins;
        this.losses = losses;
    }

    /** Question 4 (4 pts)
     *   -1 missing public keyword
     *   -1 wrong parameter types or number
     *   -2 for incorrect body
     */
    public FootballTeam (String name) {
        this(name, 0, 0);
    }

    /** Question 5 (4 pts)
     *   -2 if any method has wrong signature
     *   -1 if any method missing
     *   -1 for wrong body(s)
     */
    public String getName() { return name; }
    public int getWins() { return wins; }
    public int getLosses() { return losses; }

    /** Question 6 (4 pts)
     *   -2 if any method has wrong signature
     *   -2 for wrong body(s)
     */
    void win() { wins++; }
    void lose() { losses++; }

    /** Question 7 (4 pts)
     *   -2 for wrong signature
     *   -2 for wrong body
     */
    public boolean hasGoodRecord() {
        return (wins > losses);
    }

    /** Question 8 (6 pts)
     *   -2 for wrong signature
     *   -4 maximum for failure to follow specifications
     */
    public static void main(String[] args) {

        FootballTeam aiti = new FootballTeam("AITI", 3, 5);

        System.out.println(aiti.hasGoodRecord());

        for (int i = 0; i < 3; i++)
            aiti.win();

        System.out.println(aiti.hasGoodRecord());
    }
}

```

السؤال الخامس: (10 درجات)

(a) ماهي الغاية من التغليف Encapsulation والتجريد Abstraction ؟ (2 درجتان)

التجريد

يساعدك على تبسيط تمثيل نماذج المجال.

يُخفي التجريد التفاصيل غير ذات الصلة الموجودة في الكود.

يساعدك التجريد على تقسيم البرنامج إلى العديد من المفاهيم المستقلة.

التغليف

يساعدك التغليف على تغيير جزء من التعليمات البرمجية دون التأثير على أجزاء أخرى من التعليمات البرمجية.

يحسن قراءة التعليمات البرمجية للتطبيق

أمان محسّن ويجعل صيانة التطبيق أسهل

(b) عدل في الصنف التالي بحيث تطبق فيه خاصية التغليف على الحقول و دوال الوصول و الإسناد `a getter` `method and a setter method` لهذا الصنف . (4 درجات)

```
public class Player {  
    String name;  
    int score;  
}  
  
public class Player {  
    private String name;  
    private int score;  
    public String getName() {  
        return name;  
    }  
    public void setName(String name) {  
        this.name = name;  
    }  
    public int getScore() {  
        return score;  
    }  
    public void setScore(int score) {  
        this.score = score;  
    }  
}
```

(c) في منظومة الدراسة والامتحانات الطالب كائن يحمل الخصائص (الاسم، رقم القيد) والدوال (دالة البناء لادخال الاسم ورقم القيد، دالة حساب المتوسط لمادة $avg(g1,g2,g3)$ اي متوسط (درجة الامتحان الاول، درجة الامتحان الثاني، درجة النهائي تتم قراءتهم في البرنامج الرئيسي وترسل الى الدالة)، ودالة حساب المتوسط $avg(gArray)$ للمواد التي نجح فيها مخزنة في مصفوفة وعددها 5 يتم حجز المصفوفة وقراءتها في البرنامج الرئيسي، دالة $toString$ طباعة حالة الكائن)
في البرنامج الرئيسي احجز كائن من نوع طالب واستدعي الدالة لحساب متوسط مادة والدالة حساب متوسط مواد واطبع حالة الكائن. **(4 درجات)**

```
class Student
{
    String name;
    long id;
    Student(String nm , long ids)
    {
        this.name=nm;
        this.id=ids;
    }
    public double avg(float g1,float g2,float g3)
    {
        double sum=0.0;
        sum+=g1+g2+g3;
        return(sum/3.0);
    }
    public double avg(float [] gArray)
    {
        int n=5;
        double sum=0.0;
        for(int i=0;i<5;i++)
        {
            sum+=gArray[i];
        }
        return(sum/5.0);
    }
    public void toStr()
    {
        System.out.println("Student name: " + this.name);
        System.out.println("Student ID: " + this.id);
    }
}

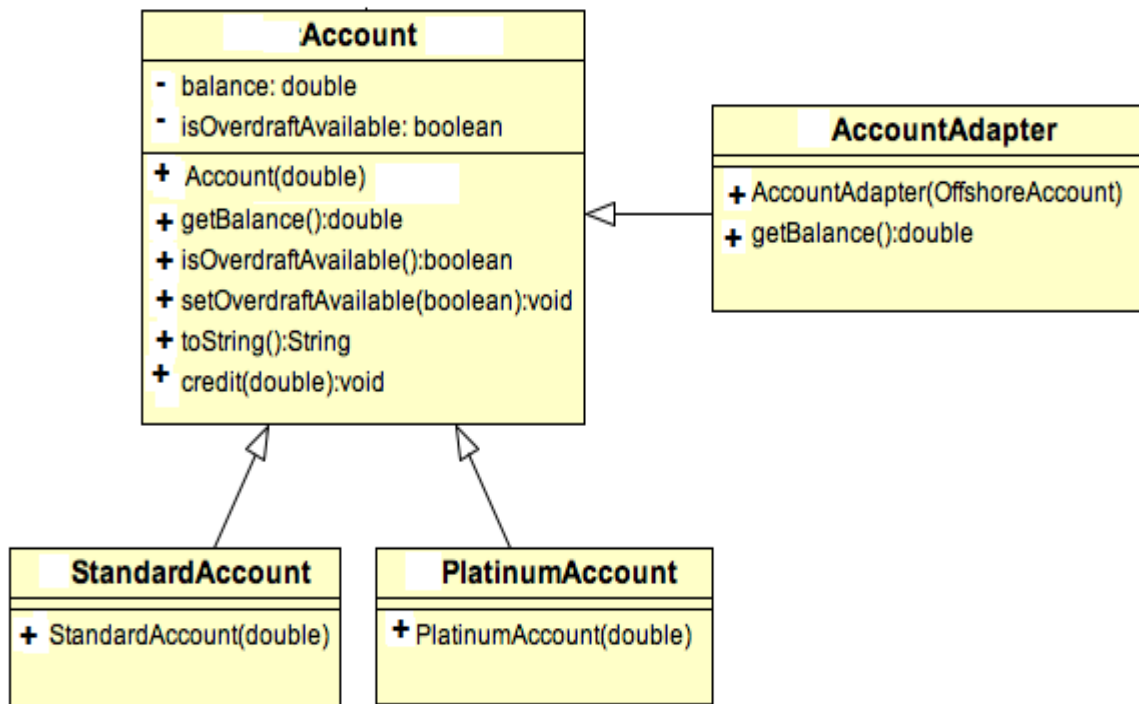
public class StudentRoll {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        String name;
        long id;
        float g1,g2,g3;
        g1=in.nextFloat();
        g2=in.nextFloat();
        g3=in.nextFloat();
    }
}
```

```

float [] gArray = new float[5];
for(int k =0;k<5;k++)
{
    gArray[k]=in.nextFloat();
}
System.out.println("Enter student name :");
name=in.next();
System.out.println("Enter student ID :");
id=in.nextLong();
Student st=new Student(name,id);
double av1=st.avg(g1, g2, g3);
double av2=st.avg(gArray);
st.toStr();
System.out.println("AVG :"+av1);
System.out.println("Total AVG :"+av2);
}
    
```

السؤال السادس: (6 درجات)

حول المخطط UML التالي إلى برنامج بلغة الجافا



```

Class Account {
Private double balance;
Private Boolean isOverdraftAvailable;
Account (double bal)
{
This.balance=bal;
}
Public double getBalance( ){ return this.balance; }
Public Boolean isOverdraftAvailable( ) { return isOverdraftAvailable; }
Public void SetOverdraftAvailable ( Boolean oda ) { this. isOverdraftAvailable = oda; }
Public string toString( void ) {
system.out.println( this.getBalance( ) + " " + isOverdraftAvailable ( ) )
}
Public void credit ( double bal) { if (this.balance != 0.0) this.balance - = bal ;}
}
Class standardAccount extends Account {
standardAccount (double bal )
{
Super (bal);
}
}
Class PlatinumAccount extends Account {
PlatinumAccount (double bal )
{
Super (bal);
}
}

Class AccountAdapter extends Account {
int offshoreAccount;
AccountAdapter (int offsAccount)
{
This. offsAccount= offshoreAccount;
}

Public double getBalance ( ) { return Super.getBalance ( ); }
}

```

انتهت الأسئلة - تمنياتنا للجميع بالتوفيق والنجاح